

**IN THE CLAIMS:**

Please insert amended Claims 1, 14, 28 and 41 as indicated hereinbelow.

Claims 37, 38 and 40 remain as original.

Claims 2-7 and 10 - 12, 13, 15 - 20, 23 - 26, and claims 30 - 33 and 39 are dependent claims which remain as previously presented. Claims 2, 8, 9, 15, 21, 22, 27, 29, 35 and 36 have been canceled.

**AMENDED CLAIMS:**

1. (currently amended) A method for automatically adapting a legacy software application that includes interface specification definitions which include definitions of screen formats, including data fields created from legacy source code and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, said method comprising the steps of:

(a) providing a software application which utilizes said legacy source code to automatically generate a series of user interface software components from the screen format definitions to provide a graphical user interface providing at least data entry and display facilities of the interface specification definitions, ~~the said graphical user interface being embodied in~~ a series of executable software components being executable by a scripting language running on said interconnected network computing resource ~~that provide the functionality for interaction with the legacy software application~~, the components containing object oriented methods for setting or obtaining values of said data fields and being executable by at least one of said computing resources in said network environment, and ~~wherein~~ upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application, wherein the executable software components are capable of interacting with one of the scripting

language or an intermediary software application to invoke user provided code.

2. (canceled)

3. (previously presented) A method in accordance with Claim 1, wherein the interface software components are arranged to generate forms corresponding to forms generated by the legacy software application.

4. (previously presented) A method in accordance with Claim 1, comprising the step of generating client interface components, the client interface components being arranged to interact over the network with the legacy software application.

5. (previously presented) A method in accordance with Claim 3, the client interface components include a user input object which is arranged to receive data input by a user of the network computing resource and transmit the data to the legacy application, over the network.

6. (previously presented) A method as claimed in Claim 1, wherein said series of software components are loadable and executable by an Internet Browser.

7. (previously presented) A method as claimed in Claim 1, wherein said series of software components comprise Java code applets.

8. (canceled)

9. (canceled)

10. (previously presented) A method as claimed in Claim 1, wherein said network environment comprises the Internet network.

11. (previously presented) A method as claimed in Claim 1, wherein said network environment utilizes TCP/IP transfer protocols.

12. (previously presented) A method as claimed in Claim 1, wherein said source code is written in 4GL language.

13. (original) A method as claimed in Claim 12 wherein said source code is written in the LINC language.

14. (currently amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for automatically adapting a legacy software application that includes interface specification definitions which include definitions of screen formats, created from legacy source code including data fields and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected

network computing resources, said method comprising the steps of:

providing a software application which utilizes said source code to automatically generate a series of user interface software components from the screen format definitions to provide a graphical user interface providing at least data entry and display facilities of the interface specification definitions, ~~the said graphical user interface being embodied in~~ a series of executable software components executable by a scripting language running on said interconnected network computing resource ~~that provides the functionality for interaction with said legacy software application,~~ said components containing objected oriented methods for setting and obtaining values of said data fields and being executable by at least one of said computing resources in said network environment, and ~~wherein~~ upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission of receipt information to and from said legacy software application, wherein the executable software components are capable of interacting with one of the scripting language or an intermediary software application to invoke user provided code.

15. (canceled)

16. (previously presented) A program storage device in accordance with Claim 11, wherein the interface software components are arranged to generate forms corresponding to forms generated by the legacy software application.

17. (previously presented) A program storage device in accordance with Claim 11, the method further comprising the step of generating client interface components, said client interface components being arranged to interact over the network with the legacy software application.

18. (original) A program storage device in accordance with Claim 17, wherein the client interface components include a user input object which is arranged to receive data input by a user of the network computing resource and transmit the data to the legacy application, over the network.

19. (previously presented) A program storage device in accordance with Claim 11, wherein said series of software components are loadable and executable by an Internet Browser.

20. (previously presented) A program storage device in accordance with Claim 11, wherein said series of software components comprise Java code applets.

21. (canceled)

22. (canceled)

23. (previously presented) A program storage device in accordance with Claim 11, wherein said network environment comprises the Internet network.

24. (previously presented) A program storage device in accordance with Claim 11, wherein said network environment utilizes TCP/IP transport protocols.

25. (previously presented) A program storage device in accordance with Claim 11, wherein said source code is written in the LINC language.

26. (previously presented) A program storage device in accordance with Claim 11, wherein said terminal screen definitions are written in a screen control language.

27. (canceled)

28. (currently amended) A system for automatically adapting a legacy software application that includes interface specification definitions which include definitions of screen formats, including data fields created from legacy source code and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, the system comprising:

means utilizing said legacy source code to automatically generate a series of user interface software components from the screen format definitions to provide a graphical user interface providing at least data entry and display facilities of the

interface specification definitions, the ~~graphical user interface being embodied in a~~ series of executable software components executable by a scripting language running said interconnected network computing resource ~~for providing the functionality for interaction with the legacy software application,~~ said executable software components containing objection oriented methods for obtaining values of said data fields and being executable by at least one of said computing resources in said network environment, and ~~wherein~~ upon execution, said computing resource is caused to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application, wherein the executable software components are capable of interacting with one of the scripting language or an intermediary software application to enable the use of user hooks permitting users to have their own code invoked without requiring any changes to the generated client code.

29. (canceled)

30. (previously presented) A system in accordance with Claim 28, wherein the interface software components are arranged to generate forms corresponding to forms generated by the legacy software application.

31. (previously presented) A system in accordance with Claim 28, the means for producing the series of software components including means for producing the series of software components including means for generating client interface components, the



client interface components being arranged to interact over the network with the legacy software application.

32. (previously presented) A system in accordance with Claim 31, the client interface components including a user input object which is arranged to receive data input by a user of the network computing resource and transmit the data to the legacy application, over the network.

33. (previously presented) A system in accordance with Claim 28, wherein said series of software components are loadable and executable by an Internet Browser.

34. (previously presented) A system in accordance with Claim 28, wherein the series of software components comprise Java code applets.

35. (canceled)

36. (canceled)

37. (original) A system in accordance with Claim 28, wherein said network environment comprises the Internet network.

38. (original) A system in accordance with Claim 28, wherein said network environment utilizes TCP/IP transfer protocols.

39. (previously presented) A system in accordance with Claim 28, wherein said source code is written in 4GL language.

40. (original) A system in accordance with Claim 28, wherein said source code is written in the LINC language.

41. (currently amended) A method for adapting a legacy software application wherein said legacy software application includes interface specification definitions and which includes definitions of screen formats created from legacy source code and developed for an environment comprising a centralized computing resource interconnected to a series of computer terminal devices, said centralized computing resource connected to a network environment, wherein said network environment comprises a system of distributed, interconnected network computing resources, said method comprising the steps of:

(a) providing a software application which utilizes said legacy source code to automatically produce a series of executable software components that provide the functionality for interact with said legacy software application including the steps of:

(a1) producing said series of executable software components which include the step of:

(a1a) generating a series of user interface software components from said screen format definitions, said user interface components being arranged for execution on the network computing resource to provide a graphical user interface

providing at least data entry and display facilities of the interface specification definitions;

(alb) enabling said series of software components to be executable by scripting languages on running said network computing resource;

(b) enabling said components to be executable by at least one of said computing resources in said network environment;

(c) upon execution, causing said computing resource to interconnect with said legacy software application over said network so as to interact with said legacy software application in the transmission or receipt of information to and from said legacy software application; and

(d) interacting, by said executable software components, with one of said scripting languages or an intermediary software application to enable the use of hooks permitting users to have their own code invoked without requiring any changes to the generated client code.